Derivazione Numerica e Simbolica

di Francesco Romani

L'argomento di questo mese è il calcolo, sia numerico che simbolico, della derivata di una funzione. I due approcci al calcolo della derivata sono sostanzialmente diversi, il calcolo numerico della derivata è un procedimento approssimato, sottoposto a severe limitazioni dal punto di vista della stabilità. La discussione permette di mostrare la potenza di Mathematica come mezzo per esposizioni didattiche approfondite. Il calcolo simbolico permette di realizzare automaticamente gli stessi procedimenti usati dallo studioso umano. Nel caso della derivata l'implementazione è agevole e priva di inconvenienti.

Derivazione Numerica

La derivata di una funzione f(x) nel punto x_0 è definita come il limite (se esiste), per h che tende a 0, del rapporto incrementale $(f(x_0+h)-f(x_0))/h$. Questa espressione fornisce un'approssimazione della derivata tanto più precisa quanto più è piccolo h. Si può stimare con precisione l'errore che si commette approssimando $f'(x_0)$ con $(f(x_0+h)-f(x_0))/h$ sviluppando l'espressione $f'(x_0) - (f(x_0+h) - f(x_0))/h$ in serie di potenze di h. Usando *Mathematica* tale sviluppo si ottiene facilmente.
Per fissare le idee scegliamo f(x) = cos x, allora f'(x) = -sin x, e sia x_0 =1.1. Indichiamo con **d** la derivata di f e con **dal** la approssimazione di f'(x) per mezzo del rapporto incrementale:

In[1]:=
```
f=Cos[x];
```
In[2]:=
```
d=D[f,x]
```
Out[2]=
```
-Sin[x]
```
In[3]:=
```
dal[h_]:=(Cos[x+h]-Cos[x])/h
```
Sviluppiamo **dal[h]-d** per ottenere i primi termini dello sviluppo dell'errore.

In[4]:=
```
Series[dal[h]-d,{h,0,3}]
```
Out[4]=

$$\frac{-(\mathrm{Cos}[x]\ h)}{2} + \frac{\mathrm{Sin}[x]\ h^2}{6} + \frac{\mathrm{Cos}[x]\ h^3}{24} + O[h]^4$$

Per valori piccoli di h, il primo termine di questa approssimazione, che dipende da h, è molto più grande degli altri, che dipendono da h^2, h^3, ... In tal caso, si usa dire che l'errore è proporzionale ad h a meno di termini di grado superiore.
Per avere un idea del comportamento, in pratica, della nostra approssimazione definiamo una funzione **L10** che calcola il logaritmo in base 10, cambiato di segno, di una quantità diversa da 0 (se l'argomento è piu piccolo di 10^{-20} viene restituto il valore 20).

In[5]:=
```
L10[t_]:=If[Abs[t]<10^-20,
            20,
            N[-Log[10,Abs[t]]]];
```
Questa funzione permette di stimare il numero di cifre significative in un risultato approssimato. Per esempio, se il valore giusto è 1.239999998 e il valore approssimato 1.239999997 la differenza vale 10^{-9} e $-\mathrm{Log}_{10}(10^{-9})$ = 8.69 sta a significare che l'approssimazione possiede circa 8 cifre decimali esatte.
Definiamo anche **err[k]** applicando **L10** all'errore del rapporto incrementale con x = 1.1 e h = 10^{-k}.

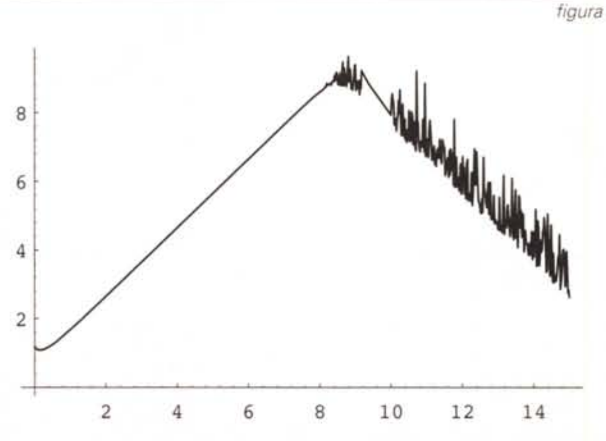
In[6]:=
```
err[k_]:=L10[(d-dal[10^-k])/.x->1.1]
```
Ciò permette di far assumere ad h valori sempre più piccoli. Stampiamo infine una tabella una tabella dei valori di **err** in funzione di h.

In[7]:=
```
Do[Print[N[10^-k],"   ",
        err[k]],{k,1,15,2}]
```

0.1	1.67415
0.001	3.64465
0.00001	5.64436
$1.\ 10^{-7}$	7.64389
$1.\ 10^{-9}$	9.51313
$1.\ 10^{-11}$	7.11598
$1.\ 10^{-13}$	5.88413
$1.\ 10^{-15}$	2.62618

Si vede come il numero di cifre di precisione dapprima salga fino a raggiungere un massimo verso h=10^{-9} per poi diminuire di nuovo. Un analisi più precisa può essere ottenuta plottando **err[k]** per k tra 0 e 15.

figura 1



In[8]:=
```
Plot[err[k], {k,0,15}];
```
La spiegazione di questo strano grafico è data dal fatto che la nostra approssimazione non è affetta soltanto dall'errore (proporzionale ad h) che abbiamo analizzato sopra e che chiameremo **errore analitico**, ma anche da un **errore di arrotondamento** che si verifica nel calcolo di $f(x_0+h)-f(x_0)$ e che, contrariamente al primo, tende a crescere al diminuire di h. Questo secondo tipo di errore, che si manifesta in special modo quando viene eseguita la sottrazione di numeri quasi uguali, può dar luogo ad un fenomeno vistoso noto come **errore di cancellazione**. Nel grafico si vede come i due errori contrapposti (analitico e di arrotondamento) si bilanciano, fornendo il migliore risultato (circa 10^{-9}) per h ~ 10^{-9}. Si nota pure che, mentre l'errore analitico ha un comportamento regolare, l'errore algoritmico è frastagliato, essendo presente una componente casuale.

Come migliorare la nostra approssimazione? Dallo sviluppo in serie, calcolato precedentemente, notiamo che come il primo termine dell'errore analitico è una funzione dispari di h. Perciò se scriviamo la stessa approssimazione sostituendo -h al posto di h, l'errore avrà valore uguale e segno opposto. Possiamo allora combinare le due approssimazioni nel modo seguente. (**Together** semplifica la funzione razionale mettendola ad uno stesso denominatore)

In[9]:=
```
Together[(da1[h]+da1[-h])/2]
```
Out[9]=
```
-Cos[h - x] + Cos[h + x]
------------------------
          2 h
```
Il primo e il terzo termine dell'errore scompaiono, dando origine ad una approssimazione più accurata.

In[10]:=
```
da2[h_]:=(Cos[x+h]-Cos[x-h])/(2h);
err2[h_]:=L10[(d-da2[10^-h])/.x->1.1];
Series[da2[h]-d,{h,0,5}]
```

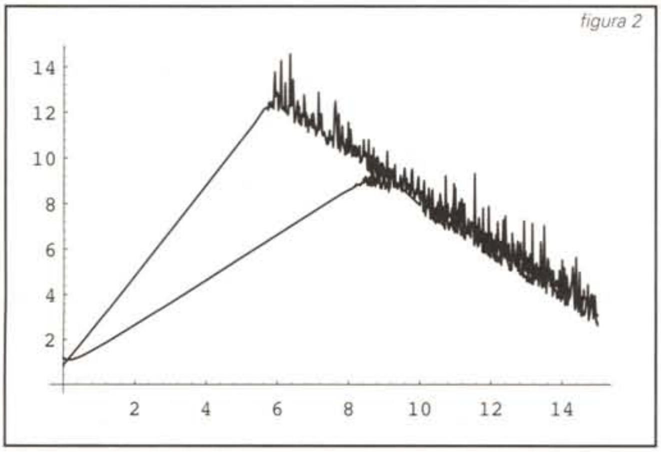


figura 2

Out[10]=
```
Sin[x] h²   Sin[x] h⁴
--------- - --------- + O[h]⁶
    6          120
```
In[11]:=
```
Plot[{err[k],err2[k]}, {k,0,15}];
```
La nuova approssimazione presenta un errore di arrotondamento che è dello stesso tipo del precedente ma siccome la pendenza dell'errore analitico è doppia (il logaritmo di h^2 è uguale a 2 volte il logaritmo di h) l'incontro tra i due grafici avviene per valori di h più grandi (circa 10^{-6}) e l'errore associato è minore (~10^{-12}).

Come migliorare ancora la nostra approssimazione? Calcoliamone un'altra con passo h/2 e sviluppiamo in serie una combinazione lineare delle due approssimazioni, con due coefficienti generici a e b.

In[12]:=
```
Series[a da2[h] + b da2[h/2] - d,
    {h,0,5}]
```
Out[12]=
```
(Sin[x] - a Sin[x] - b Sin[x]) +

  a Sin[x]   b Sin[x]
 (-------- + --------) h² +
     6          24

  -(a Sin[x])   b Sin[x]
 (----------- - --------) h⁴  + O[h]⁶
     120          1920
```
Per "uccidere" i primi due termini dello sviluppo dell'errore occorre soddisfare le condizioni 1-a+b=0 e a/6+b/24 =0. Il sistema viene risolto con la funzione **Solve** ottenendo a=-1/3 e b=4/3.

In[13]:=
```
Solve[{a+b==1,a/6+b/24==0}]
```
Out[13]=
```
           1        4
{{a -> -(-), b -> -}}
           3        3
```
Sostituendo questi valori di a e b si ha:

In[14]:=
```
Together[-1/3 da2[h] + 4/3 da2[h/2]]
```
Out[14]=
```
         h
(-8 Cos[- - x] + Cos[h - x] +
         2

          h
   8 Cos[- + x] - Cos[h + x]) / (6 h)
          2
```
L'approssimazione risultante ha errore di ordine h^4.

In[15]:=
```
da4[h_]:= -1/3 da2[h]+4/3 da2[h/2];
err4[h_]:=L10[(d-da4[10^-h])/.x->1.1];
Series[da3[h]-d,{h,0,5}]
```
Out[15]=
```
Sin[x] h⁴
--------- + O[h]⁶
   480
```
Il grafico seguente (figura 3) mostra l'andamento dei tre errori. L'ultima approssimazione presenta una pendenza dell'errore analitico quadrupla della prima: il caso ottimale si ha per h ~10^{-3} e l'errore associato è circa 10^{-15}.

In[16]:=

Errore inerente, errore analitico ed errore di arrotondamento

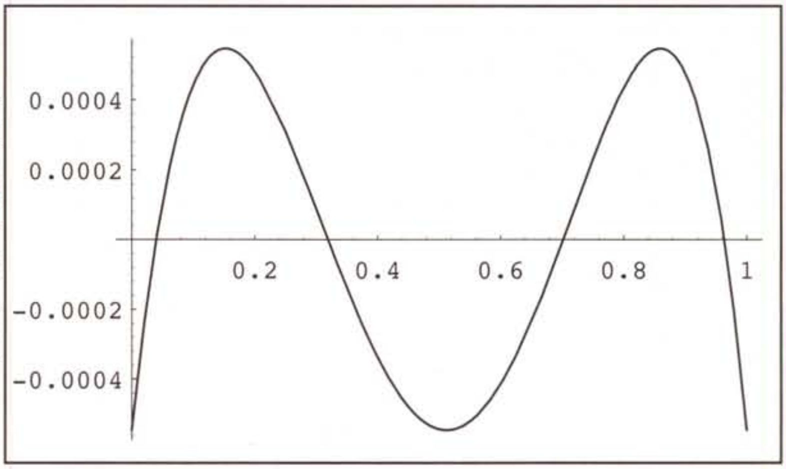
Data una funzione reale di variabile reale f(x), il suo calcolo nella aritmetica approssimata di un computer non è possibile tranne pochi casi particolari. Quella che in genere viene calcolata è una approssimazione è affetta da vari tipi di errore:

Errore inerente: Se il valore x non è rappresentabile nella aritmetica del computer o comunque è affetto da un errore Δx, quello che vine calcolato non è f(x) ma
$f(x+\Delta x) = f(x) + f'(x)\,\Delta x$ + termini di ordine superiore in Δx.

Errore analitico (o di troncamento): Se f(x) non è una funzione razionale (ovvero non è calcolabile come combinazione delle quattro operazioni aritmetiche) quello che viene calcolato non è f(x) ma una sua approssimazione razionale g(x). Per esempio, per approssimare e^x nell'intervallo [0,1] si può usare il polinomio di terzo grado (vedi Bini *et. al.* Metodi Numerici pag. 569).

$0.2799765\,x^3 + 0.421703\,x^2 + 1.016602\,x + 0.9994552$
commettendo un errore minore in modulo di $0.545\ 10^{-3}$.

In[1]:=
```
f[x_]:= 0.2799765 x^3 + 0.421703 x^2 +
        1.016602  x   + 0.9994552;
Plot[f[x] - Exp[x],{x,0,1}];
```



Errore di arrotondamento (o algoritmico): Una funzione razionale viene calcolata con una opportuna sequenza delle quattro operazioni aritmetiche; ogni operazione, effettuata in aritmetica finita è affetta da un errore. La combinazione di tutti gli errori da origine al cosiddetto errore di arrotondamento. Un esempio di errore di arrotondamento, spesso drammatico, è il fenomeno della cancellazione ovvero la perdita di cifre significative dovute alla differenza di numeri molto vicini. Calcoliamo con 19 e con 10 cifre l'espressione $y = \cos 1.1 - \cos(1.1+10^{-10})$ e confrontiamo i risultati:

```
      0.4535961214255773887 -
      0.4535961213364566516 =
y20 = 0.0000000000891207371

      0.4535961214 -
      0.4535961213 =
y10 = 0.0000000001
```

Pur operando con 19 cifre, y_{20} ha solo 9 cifre significative, y_{10} addirittura ha solo una cifra significativa, con un errore relativo di circa il 10%.
L' errore di arrotondamento è detto anche errore algoritmico perché può dipendere dal particolare algoritmo scelto per effettuare il calcolo. Per esempio, consideriamo la somma delle potenze settime dei numeri da 1 a 1000 che è un numero di 24 cifre.

In[1]:=
```
vet=Table[i^7,{i,1000}];
```

In[2]:=
```
SommaEsatta = Apply[Plus,vet]
```
Out[2]=
```
125500583333041666750000
```

Operando con la precisione di 19 cifre decimali e sommando gli addendi in ordine crescente si ottiene un errore di 8192.

In[3]:=
```
SommaCrescente = 0;
Do[SommaCrescente = SommaCrescente +
   N[vet[[i]]], {i,1000}];
SommaCrescente - SommaEsatta
```

Out[3]=
```
-8192.
```

Sommando invece in ordine decrescente si ottiene un errore 4 volte maggiore.

In[4]:=
```
SommaDecrescente = 0;
Do[SommaDecrescente =
     SommaDecrescente +
     N[vet[[i]]], {i,1000,1,-1}];
SommaDecrescente - SommaEsatta
```

Out[4]=
```
-32768
```

Il lettore è invitato a trovare da solo la motivazione. (Suggerimento: provate a fare a mano un calcolo simile per pochi addendi e con poche cifre di precisione).

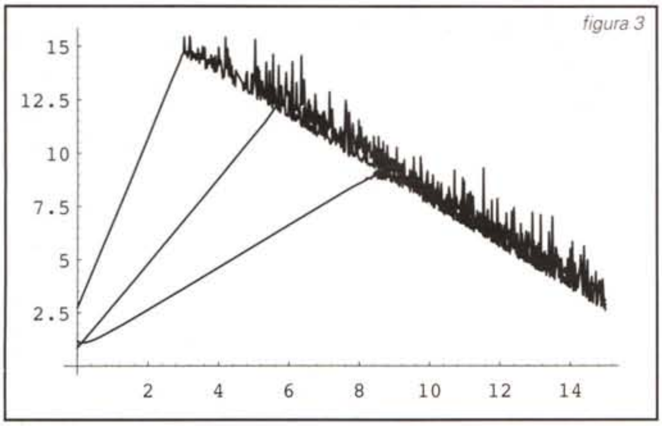


figura 3

```
Plot[{err[k],err2[k],err4[k]}, {k,0,15}];
```

Derivazione simbolica

Quando la funzione da derivare è una funzione analitica nota (come ad esempio **Exp** o **Cos**) la sua derivata è spesso conosciuta dalla teoria e può essere ottenuta consultando le tabelle. Se la funzione è formata componendo funzioni note, la derivata si ottiene combinando l'applicazione delle regole di derivazione e la consultazione delle tabelle. In *Mathematica* la derivazione simbolica si fa essenzialmente nello stesso modo, con l'uso massiccio del *pattern-matching* nelle definizioni. Vediamo di definire una funzione **Der[**<expr>**,t]** che calcola la derivata di <expr> rispetto a t (una funzione simile esiste già nel sistema ed è la **D[**<expr>**, t]**).

La prima definizione realizza la derivazione della somma:

d/dt (f(t)+g(t)) = f'(t) + g'(t).

In[1]:=

```
Der[f_+g_,t_] :=
      Der[f,t] + Der[g,t];
```

La seconda definizione realizza la derivazione del prodotto:

d/dt (f(t) g(t)) = f'(t) g(t) + f(t) g'(t).

In[2]:=

```
Der[f_ g_,t_] :=
      g Der[f,t] + f Der[g,t];
```

Si passa poi a trattare due casi specifici df/dt = 0 se f non è funzione di t; t' = 1.

In[3]:=

```
Der[f_, t_] := 0 /; FreeQ[f,t];
Der[t_, t_] := 1;
```

Le regola di derivazione di una potenza è più complicata:

$d/dt\ (f(t)^{g(t)}) = g(t)\ f(t)^{g(t)-1}\ f'(t) + g'(t)\ f(t)^{g(t)}\ Ln\ f(t)$;

si noti come tale regola si riduce alle più semplici regole per la derivazione di $f(t)^g$ e di $f^{g(t)}$ se vale rispettivamente f'=0 o g'=0.

In[4]:=

```
Der[f_^g_,t_] :=
      g f^(g-1) Der[f,t]+
      f^g Log[f] Der[g,t];
```

Infine si realizza la regola di derivazione di una funzione composta

d/dt f(g(t)) = f'(g(t)) g'(t).

In[5]:=

```
Der[f_[g_], t_] :=
   (Der[f[t],t]/.t->g) Der[g,t];
```

L'espressione tra parentesi realizza la derivata di f calcolata nel punto t=g[t], il tutto viene moltiplicato per g'(t)

Le tabelle sono implementate con una serie di definizioni dirette.

In[6]:=

```
Der[Exp[t_],t_] := Exp[t];
Der[Sin[t_],t_] := Cos[t];
Der[Cos[t_],t_] := -Sin[t];
Der[Log[t_],t_] := 1/t;
                 ...
```

Calcoliamo con la nostra funzione d/dt (cos sin t) e confrontiamo il risultato con la soluzione data da **D[]**.

In[7]:=

```
Der[Cos[Sin[x]],x]
```

Out[7]]=

```
-(Cos[x] Sin[Sin[x]])
```

In[8]:=

```
D[Cos[Sin[x]],x]
```

Out[8]=

```
-(Cos[x] Sin[Sin[x]])
```

E' doveroso notare che l'approccio presentato, pur valido dal punto di vista didattico, è largamente incompleto e abbastanza pericoloso. Se infatti si prova a calcolare la derivata di una funzione f(g(t)) in cui la derivata **Der[f[t],t]** non viene valutata, la nostra definizione *In[5]* genera un loop infinito (che il *Kernel* blocca dopo qualche centinaio di ricorsioni).

In[9]:=

```
Der[f[Sin[x]],x]

$RecursionLimit::reclim: Recursion depth of 256
exceeded.

$RecursionLimit::reclim: Recursion depth of 256
exceeded.

$RecursionLimit::reclim: Recursion depth of 256
exceeded.
```

L'approccio di *Mathematica* al calcolo delle derivate è molto più raffinato e complesso e con la **D[]** non si cade in simili errori.

In[10]:=

```
D[f[Sin[x]],x]
```

Out[10]=

```
-(Cos[x] f'[Sin[x]])
```

L'uso della **FullForm** ci mostra che c'è molta più carne al fuoco di quel che sembra.

In[11]:=

```
FullForm[f'[Sin[x]]]
```

Out[11]=

```
Derivative[1][f][Sin[x]]
```

Per comprenderne di più conviene guardare sul manuale le definizioni di **D[]** e di **Derivative[]**.

MC

Francesco Romani è raggiungibile tramite Internet all'indirizzo romani@di.unipi.it

Dove trovare di più

R.Bevilacqua, D. Bini, M. Capovani, O.Menchi. **Introduzione alla Matematica Computazionale.** Zanichelli (1987)

R.Bevilacqua, D. Bini, M. Capovani, O.Menchi. **Metodi Numerici.** Zanichelli (1992)

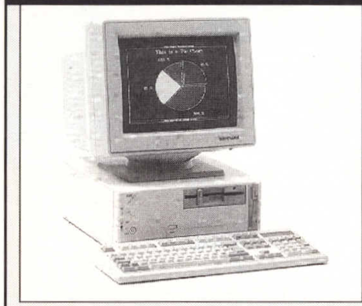
S. Wolfram. ***Mathematica.* A System for Doing Mathematics by Computer.** Addison Wesley,1991 (II Edition).

PERSONAL SELF SERVICE
SUPERMARKET DELL'INFORMATICA

VENDITA - PERMUTE - NOLEGGIO PC
ASSEMBLATI NUOVI E USATI - SPEDIZIONI
POSTALI IN TUTTA ITALIA - ASSISTENZA TECNICA

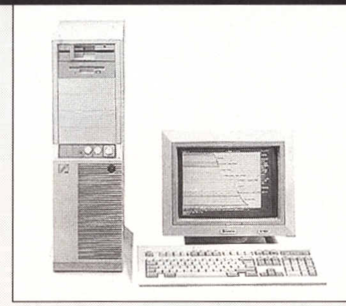
WIN COMPUTER

PERSONAL COMPUTER PC WIN



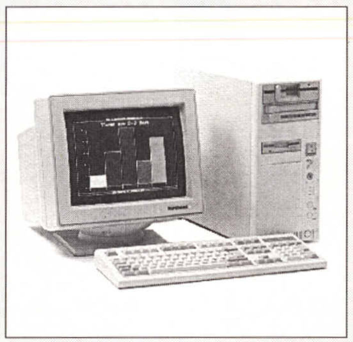
WIN COMPUTER 486DX/33
BOARD 486DX/33 LOCAL BUS, 256K C., 2MB RAM, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER + 1 PAR, VGA 1 MB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 2.190.000

WIN COMPUTER 486DX2/50
BOARD 486DX2/50, LOCAL BUS, 256K C., 2MB RAM, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER + 1 PAR, VGA 1 MB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 2.390.000



WIN COMPUTER 386SX/40
BOARD 386SX/40, 2MB RAM, CASE DESKTOP, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER + 1 PAR, VGA 256 KB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 1.390.000

WIN COMPUTER 386DX/40
BOARD 386DX/40, 128K C., 2MB RAM, CASE DESKTOP, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER 1 PAR, VGA 1 MB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 1.590.000



WIN COMPUTER 486DX/50
BOARD 486DX/50 LOCAL BUS, 256K C., 2MB RAM, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER + 1 PAR, VGA 1 MB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 2.490.000

WIN COMPUTER 486DX2/66
BOARD 486DX2/66, LOCAL BUS, 256K C., 2MB RAM, FLOPPY 1.44 MB, HARD DISK 130MB, MULTI I/O IDE 2 SER + 1 PAR, VGA 1 MB, MONITOR 14" SVGA COLORI, TASTIERA.
LIRE 2.690.000

PRODOTTI MULTIMEDIALI

SOUNDMED ADLIB COMPAT.	90.000
SOUND BLASTER PRO BASIC	275.000
SOUND BLASTER 16 ASP	398.000
KIT INTERFACCIA MIDI	80.000
MIDI BLASTER	390.000
ADATTATORE VGA TO PAL	198.000

VIDEO BLASTER 495.000
EFFETTI VIDEO DIGITALI, WINDOWS 3.1 COMPATIBILE, AMPLIFICATORE, MIXER STEREO, PCX, TIFF, BMP, ECC.

SOUND B. MULTIMEDIA KIT 990.000
BUSINESS KIT CONTIENE SCHEDA PRO, 2 CASSE LETTORE CD ROM INTERNO, 10 TITOLI CD

CD ROM MITSUMI 430.000
MONTAGGIO INTERNO, 350ms, JACK RCA PER CUFFIE VOLUME, CD KODAK, COMP. SOUNDBLASTER, CTRL 16 BIT

TV VIDEO MAGIC 690.000
LA TUA TV IN AMBIENTE WINDOWS, 100 CANALI MEMORIZZABILI INGRESSO TELECAMERA O VIDEOREGISTRATORE USCITA STEREO

PRODOTTI INTEGRATIVI PER PC

BOARD

286/20		140.000
386SX/33		190.000
386SX/40		190.000
386DX/40	128K CACHE	290.000
486DX/33	LOCAL BUS	850.000
486DX2/50	LOCAL BUS	1.100.000
486DX/50	LOCAL BUS	1.200.000
486DX2/66	LOCAL BUS	1.400.000

HARD DISK

HD	43Mb	3,5"	250.000
HD	130Mb	3,5"	350.000
HD	210Mb	3,5"	490.000
HD	250Mb	3,5"	590.000
HD	340Mb	3,5"	790.000
HD	452Mb	3,5"	1.090.000
HD	85Mb	2,5"	1.090.000
HD	130Mb	2,5"	440.000
HD	200Mb	2,5"	590.000

MONITOR

14"	VGA monocr.		210.000
14"	VGA col.	800	390.000
14"	SVGA col.	1024	490.000
14"	SVGA col.	1024 LR	550.000
14"	SVGA col.	1280 NI	590.000

STAMPANTI HP

LASERJET 4L	1.499.000
LASERJET 4	3.250.000
DESKJET 510	749.000
DESKJET 550C	1.490.000
DESKJET 1200C	2.980.000

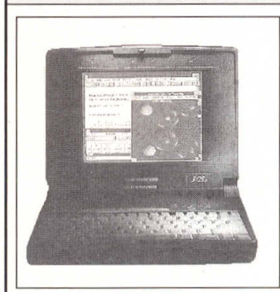
STAMPANTI EPSON

STYLUS 800	590.000
LX 400	320.000
LQ 100	390.000
EPL 5000	1.090.000

STAMPANTI CITIZEN

120 D+	230.000
SWIFT 9	420.000
SWIFT 9X	590.000
SWIFT 200	550.000
SWIFT 240	650.000
SWIFT 240 COLORE	690.000
SWIFT 24X	860.000
PN 48 PORTATILE	520.000

SPECIALE NOTEBOOK

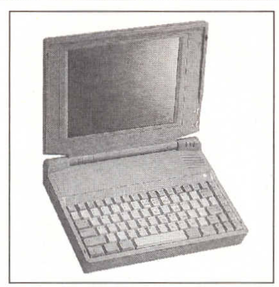


NOTEBOOK CONTURA COMPAQ 3/25
CPU 386SLX/25, 4 MB RAM, ESP. A 12, FDD 1,44 MB, HD 80 MB, LCD 10", TRACK-BALL, DOS 5.0, WINDOWS 3.1
LIRE 2.860.000

NOTEBOOK CONTURA COMPAQ 3/25C
CPU 386SLX/25, 4 MB RAM, FDD 1,44 MB, HD 80 MB, LCD 10", COLORE TRACK-BALL, DOS 5.0, WINDOWS 3.1
LIRE 3.750.000

NOTEBOOK NP 204 486SLC/25-33
CPU 486SLC/25, 4 MB RAM, FLOPPY 1,44, HD 80 MB, DISPLAY 10" VGA 64 T. GRIGIO, BORSA, KG. 2,9
LIRE 2.390.000

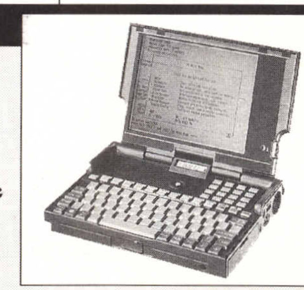
NOTEBOOK NP 204 486SLC/25-33
CPU 486SLC/25, 4 MB RAM, FLOPPY 1,44, HD 120 MB, DISPLAY 10" VGA 64 T. GRIGIO, BORSA, KG. 2,9
LIRE 2.590.000



PC Quaderno Olivetti

CPU NEC V30HL 8086 16 Mhz, 1 MB RAM, Hard Disk 20MB, lettore IC-CARD 1.0 Video display D CGA integrata 640x400 con 8 toni di grigio, MS-DOS 5.0, Interlink, note, calcolatrice, agenda, rubrica, gestore file, registratore audio sistema voice processing microfono + altoparlante incorporato, peso gr. 1050

LIRE 750.000



VIA MATERA, 3 - 00182 ROMA
TEL. (06) 702.58.94/45.44/45.32
FAX (06) 70.47.67.15
(ZONA SAN GIOVANNI)
FERMATA METRO RE DI ROMA

VIA L. ZAMBARELLI, 16 - 00152 ROMA
TEL. (06) 58201066 - 58201067
FAX (06) 58201067
(ZONA MONTEVERDE)
FERMATA S. GIOVANNI DI DIO

ORARIO: 9.00/13.00 - 15.00/19.30 - SABATO MATTINA APERTO - I PREZZI SOPRA INDICATI SONO DA INTENDERSI IVA E MONTAGGIO ESCLUSI - IL PRESENTE LISTINO HA VALIDITÀ PER CAMBIO DOLLARO MASSIMO DI LIRE 1.400

HAPPY CARD

L'INIZIATIVA È RISERVATA AGLI ACQUIRENTI DEI PERSONAL COMPUTER PC WIN AI QUALI VERRÀ CONSEGNATA GRATUITAMENTE UNA CARD CHE OFFRE I SEGUENTI SERVIZI: 12 MESI DI GARANZIA SU TUTTI I PRODOTTI, RIPARAZIONE ENTRO LE 24 ORE, HOT LINE TELEFONICA HARDWARE E SOFTWARE, LIBERO ACCESSO A FREE BITS BBS, SCONTI PARTICOLARI SUI SUCCESSIVI ACQUISTI.